

# Programming Assignment 3: SDN and Ryu

Student Name Avishek Mukhopadhyay

Entry No 2021MCS2125

## Contents

### 1 Prerequisite

I have used ubuntu-18.04 as my operating system. I had installed mininet and ryu for this assignment. python and ryu-mnager have been used for running

### 2 Part 1(Controller Hub) How to run

Open a terminal and run ryu-manager controllerhub.py Then open another terminal and change to root user and run python mininetpart1.py

### 3 Part 1(Controller Hub) Design strategy

The Controller hub always broadcasts all the packets it receives at each port. Similarly my code contained at controller hub.py contains a class ControllerHub. It has method called function for switch. It configures the behaviour of the switch. It also has function for packet method it is invoked upon receiving any packet. It receives the packet and broadcast it to its all ports.

### 4 Part 1(Controller Hub) Ping and Throughput

In controller hub the ping taking higher time and the throughput is less. Whereas in learning switch, first packet takes higher time but after that the ping time gets low because the switch stops broadcasting for the same port as it has learned the port vs node mapping. The throughput in learning switch is also higher beacuse lack of flooding the same message.

### 5 Part 1(Controller Hub) Output

I have executed ping from host1 to host2, 5 times with 0.5 sec gap following is the output

PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data. 64 bytes from 10.0.0.2: icmp seq=1 ttl=64 time=20.2 ms 64 bytes from 10.0.0.2: icmp seq=2 ttl=64 time=7.82 ms 64 bytes from 10.0.0.2: icmp seq=3 ttl=64 time=10.8 ms 64 bytes from 10.0.0.2: icmp seq=4 ttl=64 time=7.69 ms 64 bytes from 10.0.0.2: icmp seq=5 ttl=64 time=5.19 ms  
— 10.0.0.2 ping statistics — 5 packets transmitted, 5 received, 0percent packet loss, time 2004ms rtt min avg max mdev = 5.196 10.370 20.279 5.270 ms

```
Activities Terminal Sun 21:11
cdot@cdot-Latitude-3580: ~/Downloads/2021MCS2125
File Edit View Search Terminal Help
cdot@cdot-Latitude-3580:~/Downloads/2021MCS2125$ ryu-manager controller_hub.py
loading app controller_hub.py
loading app ryu.controller.ofp_handler
instantiating app controller_hub.py of ControllerHub
instantiating app ryu.controller.ofp_handler of OFPHandler
I have found packet with 0000000000000001 3a:d5:d3:2a:6b:07 33:33:00:00:00:16 4
I have found packet with 0000000000000001 00:00:00:00:10:11 33:33:ff:00:10:11 1
I have found packet with 0000000000000002 00:00:00:00:10:11 33:33:ff:00:10:11 1
I have found packet with 0000000000000002 00:00:00:00:10:14 33:33:00:00:00:16 2
I have found packet with 0000000000000001 00:00:00:00:10:14 33:33:00:00:00:16 4
I have found packet with 0000000000000002 00:00:00:00:10:14 33:33:ff:00:10:14 2
I have found packet with 0000000000000001 00:00:00:00:10:14 33:33:ff:00:10:14 4
I have found packet with 0000000000000002 0a:91:dd:3a:db:c3 33:33:00:00:00:16 1
I have found packet with 0000000000000002 00:00:00:00:10:15 33:33:00:00:00:16 3
I have found packet with 0000000000000001 00:00:00:00:10:15 33:33:00:00:00:16 4
I have found packet with 0000000000000002 00:00:00:00:10:15 33:33:ff:00:10:15 3
I have found packet with 0000000000000001 00:00:00:00:10:15 33:33:ff:00:10:15 4
I have found packet with 0000000000000001 3a:d5:d3:2a:6b:07 33:33:ff:2a:6b:07 4
I have found packet with 0000000000000001 00:00:00:00:10:11 33:33:00:00:00:16 1
I have found packet with 0000000000000002 0a:91:dd:3a:db:c3 33:33:ff:3a:db:c3 1
I have found packet with 0000000000000002 00:00:00:00:10:11 33:33:00:00:00:16 1
I have found packet with 0000000000000001 00:00:00:00:10:12 33:33:ff:00:10:12 2
I have found packet with 0000000000000001 00:00:00:00:10:13 33:33:ff:00:10:13 3
I have found packet with 0000000000000002 00:00:00:00:10:12 33:33:ff:00:10:12 1
I have found packet with 0000000000000002 00:00:00:00:10:13 33:33:ff:00:10:13 1
I have found packet with 0000000000000001 00:00:00:00:10:11 33:33:00:00:00:02 1
I have found packet with 0000000000000001 00:00:00:00:10:11 33:33:00:00:00:16 1
I have found packet with 0000000000000002 00:00:00:00:10:11 33:33:00:00:00:02 1
I have found packet with 0000000000000002 00:00:00:00:10:11 33:33:00:00:00:16 1
I have found packet with 0000000000000002 00:00:00:00:10:14 33:33:00:00:00:16 2
I have found packet with 0000000000000002 00:00:00:00:10:14 33:33:00:00:00:02 2
I have found packet with 0000000000000001 00:00:00:00:10:14 33:33:00:00:00:16 4

root@cdot-Latitude-3580:/home/cdot/Downloads/2021MCS2125
File Edit View Search Terminal Help
root@cdot-Latitude-3580:/home/cdot/Downloads/2021MCS2125$ python mininet_part1.py
*** Starting controller
Remotel
*** Starting 2 switches
switch1 switch2 ...
Starting My Network....
Running PINGALL ....
ping -l 0.5 -c 5 10.0.0.2 > ping.txt
*** Stopping 1 controllers
Remotel
*** Stopping 6 links
.....
*** Stopping 2 switches
switch1 switch2
*** Stopping 5 hosts
host1 host2 host3 host4 host5
*** Done
root@cdot-Latitude-3580:/home/cdot/Downloads/2021MCS2125#

ping.txt [Read-Only]
PING 10.0.0.2 (10.0.0.2): 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=20.2 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=7.82 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=7.69 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=5.19 ms

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 2004ms
rtt min/avg/max/ndev = 5.190/10.370/20.279/5.270 ms
```

# Programming Assignment 3: SDN and Ryu(Part 1 Learning Switch)

Student Name Avishek Mukhopadhyay

Entry No 2021MCS2125

## Contents

### 1 Part 1(Learning Switch) How to run

Open a terminal and run `ryu-manager leraningswitch.py` Then open another terminal and change to root user and run `python mininetpart1.py`

### 2 Part 1(Learning Switch) Design strategy

The Learning Switch initially broadcasts the packets for an unknown destination packet it receives first time and then it saves the mac to port mapping and then from then onwards it unicasts packets based on mac and port mapping . Similarly my code contained at `learningswitch.py` contains a class `LearningSwitch`. It has method called `functionforswitch`. It configures the behaviour of the switch. It also has `functionforpacket` method it is invoked upon receiving any packet. It receives the packet and broadcast first time to its all ports and then it stores the mac to port mapping and now on it unicasts for that mac id.

### 3 Part 1(Learning Switch) Ping and Throughput

In Learning Switch first ping taking higher time and the throughput is less because first time it is broadcasting. Whereas afterwards after that the ping time gets low because the switch stops broadcasting for the same port as it has learned the port vs node mapping. The throughput in learning switch is also higher beacuse lack of flooding the same message.

## 4 Part 1(Learning Switch) Output

I have executed ping from host1 to host2, 5 times with 0.5 sec gap following is the output

```
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data. 64 bytes from 10.0.0.2:
icmpseq=1 ttl=64 time=23.1 ms 64 bytes from 10.0.0.2: icmpseq=2 ttl=64
time=0.529 ms 64 bytes from 10.0.0.2: icmpseq=3 ttl=64 time=0.085 ms 64
bytes from 10.0.0.2: icmpseq=4 ttl=64 time=0.113 ms 64 bytes from 10.0.0.2:
icmpseq=5 ttl=64 time=0.083 ms
```

```
— 10.0.0.2 ping statistics — 5 packets transmitted, 5 received, 0 percent
packet loss, time 2041ms rtt min avg max mdev = 0.083 4.787 23.127 9.171 ms
```

```
Activities Text Editor Sun 21:14
cdot@cdot-Latitude-3580: ~/Downloads/2021MCS2125
File Edit View Search Terminal Help
cdot@cdot-Latitude-3580:~/Downloads/2021MCS2125$ ryu-manager learning_switch.py
loading app learning_switch.py
loading app ryu.controller.ofp.handler
instantiating app learning_switch.py of LearningSwitch
instantiating app ryu.controller.ofp.handler of OFPHandler
I have found packet with 00:00:00:00:00:0001 00:00:00:00:10:13 33:33:00:00:00:16 3
I have found packet with 00:00:00:00:00:0002 00:00:00:00:10:13 33:33:00:00:00:16 1
I have found packet with 00:00:00:00:00:0001 00:00:00:00:10:12 33:33:ff:00:00:12 2
I have found packet with 00:00:00:00:00:0002 00:00:00:00:10:12 33:33:ff:00:00:12 1
I have found packet with 00:00:00:00:00:0002 00:00:00:00:10:15 33:33:ff:00:00:15 3
I have found packet with 00:00:00:00:00:0001 00:00:00:00:10:15 33:33:ff:00:00:15 4
I have found packet with 00:00:00:00:00:0002 00:00:00:00:10:15 33:33:00:00:00:16 3
I have found packet with 00:00:00:00:00:0001 00:00:00:00:10:15 33:33:00:00:00:16 4
I have found packet with 00:00:00:00:00:0001 3e:13:f4:11:75:92 33:33:00:00:00:16 4
I have found packet with 00:00:00:00:00:0002 ce:db:75:6c:cc:cc 33:33:00:00:00:16 1
I have found packet with 00:00:00:00:00:0001 00:00:00:00:10:13 33:33:ff:00:00:13 3
I have found packet with 00:00:00:00:00:0002 00:00:00:00:10:13 33:33:ff:00:00:13 1
I have found packet with 00:00:00:00:00:0002 00:00:00:00:10:14 33:33:ff:00:00:14 2
I have found packet with 00:00:00:00:00:0001 00:00:00:00:10:14 33:33:ff:00:00:14 4
I have found packet with 00:00:00:00:00:0001 00:00:00:00:10:11 33:33:ff:00:00:11 1
I have found packet with 00:00:00:00:00:0002 00:00:00:00:10:11 33:33:ff:00:00:11 1
I have found packet with 00:00:00:00:00:0001 00:00:00:00:10:12 33:33:00:00:00:16 2
I have found packet with 00:00:00:00:00:0002 00:00:00:00:10:12 33:33:00:00:00:16 1
I have found packet with 00:00:00:00:00:0001 00:00:00:00:10:11 33:33:00:00:00:16 1
I have found packet with 00:00:00:00:00:0002 00:00:00:00:10:11 33:33:00:00:00:16 1
I have found packet with 00:00:00:00:00:0001 3e:13:f4:11:75:92 33:33:ff:11:75:92 4
I have found packet with 00:00:00:00:00:0002 00:00:00:00:10:14 33:33:00:00:00:16 2
I have found packet with 00:00:00:00:00:0001 00:00:00:00:10:14 33:33:00:00:00:16 4
I have found packet with 00:00:00:00:00:0002 ce:db:75:6c:cc:cc 33:33:00:00:00:16 1
I have found packet with 00:00:00:00:00:0002 ce:db:75:6c:cc:cc 33:33:00:00:00:02 1
I have found packet with 00:00:00:00:00:0002 ce:db:75:6c:cc:cc 33:33:00:00:00:16 1
I have found packet with 00:00:00:00:00:0001 00:00:00:00:10:12 33:33:00:00:00:16 2

root@cdot-Latitude-3580:/home/cdot/Downloads/2021MCS2125$ python mininet_part1.py
File Edit View Search Terminal Help
root@cdot-Latitude-3580:/home/cdot/Downloads/2021MCS2125# python mininet_part1.py
*** Starting controller
Remotel
*** Starting 2 switches
switch1 switch2 ...
Starting My Network....
Running PINGALL ....
ping -i 0.5 -c 5 10.0.0.2 > ping.txt
*** Stopping 1 controllers
Remotel
*** Stopping 6 links
.....
*** Stopping 2 switches
switch1 switch2
*** Stopping 5 hosts
host1 host2 host3 host4 hosts
*** Done
root@cdot-Latitude-3580:/home/cdot/Downloads/2021MCS2125#

part1_1.txt part1_2.txt ping.txt
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=23.1 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.529 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.085 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.113 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.083 ms

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 264ms
rtt min/avg/max/mdev = 0.083/4.787/23.127/9.171 ms
```

# Programming Assignment 3: SDN and Ryu(Part 2 Firewall Monitoring)

Student Name Avishek Mukhopadhyay

Entry No 2021MCS2125

## Contents

### 1 Part 2(Firewall Monitoring) How to run

Open a terminal and run `ryu-manager firewall monitoring.py` Then open another terminal and change to root user and run `python mininet part2.py`

### 2 Part 2(Firewall Monitoring) Design strategy

The Firewall Switch initially apply all the rules intended for the packets it receives. Whenever it receives any packet and checks the action list and take appropriate action. Similarly my code contained at `firewall monitoring.py` contains a class `FirewallSwitch`. It has method called `function` for switch. It configures the behaviour of the switch. It also has `function` for packet method it is invoked upon receiving any packet. Upon receiving any packet, if the switch finds that the traffic is not allowed then it drops the packet..

### 3 Part 2(Firewall Monitoring) Rules

For all combination of source and destination macm, firewall rules are applied. The number of rules can be decreased by using pattern based or regular expression based rule support.

### 4 Part 2(Firewall Monitoring) Output

I have executed ping from host1 to host2, 5 times with 0.5 sec gap This ping is successful because host1 and host2 are allowed to communicate. following is the output

```
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data. 64 bytes from 10.0.0.2: icmp seq=1 ttl=64 time=29.6 ms 64 bytes from 10.0.0.2: icmp seq=2 ttl=64
```

time=0.350 ms 64 bytes from 10.0.0.2: icmp seq=3 ttl=64 time=0.090 ms 64 bytes from 10.0.0.2: icmp seq=4 ttl=64 time=0.080 ms 64 bytes from 10.0.0.2: icmp seq=5 ttl=64 time=0.054 ms

— 10.0.0.2 ping statistics — 5 packets transmitted, 5 received, 0 percent packet loss, time 2031ms rtt min avg max mdev = 0.054 6.035 29.601 11.783 ms

Then I have executed ping from host2 to host5, 5 times with 0.5 sec gap This ping is unsuccessful because host2 and host5 are not allowed to communicate.

following is the output

PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data. From 10.0.0.2 icmp seq=1 Destination Host Unreachable From 10.0.0.2 icmp seq=2 Destination Host Unreachable From 10.0.0.2 icmp seq=3 Destination Host Unreachable From 10.0.0.2 icmp seq=4 Destination Host Unreachable From 10.0.0.2 icmp seq=5 Destination Host Unreachable

Since host2 and host5 has firewall blocking, hence ping has been blocked



Sun 21:26

cdot@cdot-Latitude-3580: ~/Downloads/2021MCS2125

cdot@cdot-Latitude-3580:~/Downloads/2021MCS2125\$ ryu-manager firewall\_monitor.py  
loading app firewall\_monitor.py  
loading app ryu.controller.ofp\_handler  
instantiating app firewall\_monitor.py of LearningSwitch  
instantiating app ryu.controller.ofp\_handler of OFPHandler  
Monitor H3 Generated Captured Packet with 0000000000000001 00:00:00:00:10:13 33:33:00:00:00:00  
16 3  
Monitor H3 Generated Captured Packet with 0000000000000002 00:00:00:00:10:13 33:33:00:00:00:00  
16 1  
Monitor H3 Generated Captured Packet with 0000000000000001 00:00:00:00:10:13 33:33:ff:00:10:10  
13 3  
Monitor H3 Generated Captured Packet with 0000000000000002 00:00:00:00:10:13 33:33:ff:00:10:10  
13 1  
Monitor H3 Generated Captured Packet with 0000000000000001 00:00:00:00:10:13 33:33:00:00:00:00  
02 3  
Monitor H3 Generated Captured Packet with 0000000000000001 00:00:00:00:10:13 33:33:00:00:00:00  
16 3  
Monitor H3 Generated Captured Packet with 0000000000000002 00:00:00:00:10:13 33:33:00:00:00:00  
02 1  
Monitor H3 Generated Captured Packet with 0000000000000002 00:00:00:00:10:13 33:33:00:00:00:00  
16 1  
Monitor H3 Generated Captured Packet with 0000000000000001 00:00:00:00:10:13 33:33:00:00:00:00  
16 3  
Monitor H3 Generated Captured Packet with 0000000000000002 00:00:00:00:10:13 33:33:00:00:00:00  
02 3  
Monitor H3 Generated Captured Packet with 0000000000000001 00:00:00:00:10:13 33:33:00:00:00:00  
02 1  
Monitor H3 Generated Captured Packet with 0000000000000001 00:00:00:00:10:13 33:33:00:00:00:00  
02 3  
Monitor H3 Generated Captured Packet with 0000000000000002 00:00:00:00:10:13 33:33:00:00:00:00

root@cdot-Latitude-3580: /home/cdot/Downloads/2021MCS2125

root@cdot-Latitude-3580:/home/cdot/Downloads/2021MCS2125\$ python mininet\_part2.py  
\*\*\* Starting controller  
Remotel  
\*\*\* Starting 2 switches  
switch1 switch2 ...  
Starting My Network....  
Running Ping from host1 ....  
ping -i 0.5 -c 5 10.0.0.2 > ping2.txt  
Running Ping from host2 ....  
ping -i 0.5 -c 5 10.0.0.5 >> ping2.txt  
\*\*\* Stopping 1 controllers  
Remotel  
\*\*\* Stopping 6 links  
.....  
\*\*\* Stopping 2 switches  
switch1 switch2  
\*\*\* Stopping 5 hosts  
host1 host2 host3 host4 host5  
\*\*\* Done  
root@cdot-Latitude-3580:/home/cdot/Downloads/2021MCS2125#

mininet\_part2.py

firewall\_monitor.py

ping2.txt

Text Editor

PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:  
64 bytes from 10.0.0.2: icmp\_seq=1 ttl=64 time=29.6 ms  
64 bytes from 10.0.0.2: icmp\_seq=2 ttl=64 time=0.350 ms  
64 bytes from 10.0.0.2: icmp\_seq=3 ttl=64 time=0.090 ms  
64 bytes from 10.0.0.2: icmp\_seq=4 ttl=64 time=0.080 ms  
64 bytes from 10.0.0.2: icmp\_seq=5 ttl=64 time=0.854 ms  
  
--- 10.0.0.2 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 203ms  
rtt min/avg/max/mdev = 0.054/0.035/29.601/11.783 ms  
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data:  
From 10.0.0.2 icmp\_seq=1 Destination Host Unreachable  
From 10.0.0.2 icmp\_seq=2 Destination Host Unreachable  
From 10.0.0.2 icmp\_seq=3 Destination Host Unreachable  
From 10.0.0.2 icmp\_seq=4 Destination Host Unreachable  
From 10.0.0.2 icmp\_seq=5 Destination Host Unreachable  
  
--- 10.0.0.5 ping statistics ---  
5 packets transmitted, 0 received, 100% packet loss, time 206ms  
pipe 5

Alt selected (909 bytes)

The image shows a Linux desktop environment with a dark theme. On the left is a vertical dock containing icons for various applications including a file manager, web browser, and terminal. The main workspace features three terminal windows. The top-left terminal window displays the output of a script named 'firewall\_monitor.py', which shows multiple 'Monitor H3 Generated Captured Packet' entries with hexadecimal data. The top-right terminal window shows the execution of 'python mininet\_part2.py', which starts and stops a Mininet network simulation, including commands like 'Starting controller', 'Starting 2 switches', and 'Stopping 1 controllers'. The bottom terminal window shows the output of a ping test, displaying statistics for pinging 10.0.0.2 and 10.0.0.5, including packet loss percentages and round-trip times. The desktop background is a solid dark color, and the overall layout is clean and organized.

# Programming Assignment 3: SDN and Ryu(Part 3 Load Balancer)

Student Name Avishek Mukhopadhyay

Entry No 2021MCS2125

## Contents

### 1 Part 3(Load Balancer) How to run

Open a terminal and run ryu-manager load balancer.py Then open another terminal and change to root user and run python mininet part3.py

### 2 Part 3(Load Balancer) Design strategy

The Load Balancer distributes load in round robin fashion among target hosts. Whenever it receives any packet and checks last selected target and take a new target and send the packet toward it. Similarly my code contained at load balancer.py contains a class LoadBalancing. It has method called function for switch. It configures the behaviour of the switch. It also has function for packet method it is invoked upon receiving any packet. Upon receiving any packet, if switch distributes the packet among host4 and host5.

### 3 Part 3(Load Balancer) Output

I have executed ping from host1 and host2 to 10.0.0.42, Each time a different target host was selected i.e. first time 10.0.0.4 and second time 10.0.0.5 was selected following is the output

Data Packet from ip: 10.0.0.1 Data Packet dest ip : 10.0.0.42 Selected Target Server ip is 10.0.0.4 Data Packet sent to : 10.0.0.4

Sending Reply from: 10.0.0.4 Sending reply to 10.0.0.1

Data Packet from ip: 10.0.0.2 Data Packet dest ip : 10.0.0.42 Selected Target Server ip is 10.0.0.5 Data Packet sent to : 10.0.0.5

Sending Reply from: 10.0.0.5 Sending reply to 10.0.0.2

Activities Terminal Sun 21:51

cdot@cdot-Latitude-3580: ~/Downloads/2021MCS2125

File Edit View Search Terminal Help  
cdot@cdot-Latitude-3580:~/Downloads/2021MCS2125\$ ryu-manager load\_balancer.py  
loading app load\_balancer.py  
loading app ryu.controller.ofp\_handler  
Instantiating app load\_balancer.py of LoadBalancing  
Instantiating app ryu.controller.ofp\_handler of OFPHandler  
  
Data Packet from ip: 10.0.0.1  
Data Packet dest ip : 10.0.0.42  
Selected Target Server ip is 10.0.0.4  
Data Packet sent to : 10.0.0.4  
  
Sending Reply from: 10.0.0.4  
Sending reply to 10.0.0.1  
  
Data Packet from ip: 10.0.0.2  
Data Packet dest ip : 10.0.0.42  
Selected Target Server ip is 10.0.0.5  
Data Packet sent to : 10.0.0.5  
  
Sending Reply from: 10.0.0.5  
Sending reply to 10.0.0.2  
  
Data Packet from ip: 10.0.0.42  
Data Packet dest ip : 10.0.0.42  
Selected Target Server ip is 10.0.0.4  
Data Packet sent to : 10.0.0.4  
  
Sending Reply from: 10.0.0.4  
Sending reply to 10.0.0.42  
  
Data Packet from ip: 10.0.0.42  
Data Packet dest ip : 10.0.0.42  
Selected Target Server ip is 10.0.0.5  
Data Packet sent to : 10.0.0.5  
  
Sending Reply from: 10.0.0.5  
Sending reply to 10.0.0.42  
█

root@cdot-Latitude-3580: /home/cdot/Downloads/2021MCS2125

File Edit View Search Terminal Help  
root@cdot-Latitude-3580:/home/cdot/Downloads/2021MCS2125# python mininet\_part3.py  
\*\*\* Starting controller  
Remote1  
\*\*\* Starting 2 switches  
switch1 switch2 ...  
Starting My Network....  
Running Ping from host1 ....  
ping -t 1 -c 5 10.0.0.42  
Running Ping from host2 ....  
ping -t 1 -c 5 10.0.0.42  
█